(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 633 949 A1

(12) EUROPEAN PATENT APPLICATION
published in accordance with Art. 153(4) EPC

(43) Date of publication:
08.04.2020 Bulletin 2020/15

(21) Application number: **18910272.6**

(22) Date of filing: **28.05.2018**

(51) Int Cl.:
H04L 29/06 *(2006.01)* ***H04L 29/08*** *(2006.01)*

(86) International application number:
PCT/CN2018/088676

(87) International publication number:
WO 2019/178942 (26.09.2019 Gazette 2019/39)

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR
Designated Extension States:
BA ME
Designated Validation States:
KH MA MD TN

(30) Priority: **23.03.2018 CN 201810247194**

(71) Applicant: **Wangsu Science & Technology Co., Ltd.**
Shanghai 200030 (CN)

(72) Inventors:
- **LIN, Jinpeng**
  Shanghai 200030 (CN)
- **WANG, Wencan**
  Shanghai 200030 (CN)
- **DONG, Shujia**
  Shanghai 200030 (CN)

(74) Representative: **Vinsome, Rex Martin**
Urquhart-Dykes & Lord LLP
12th Floor
Cale Cross House
156 Pilgrim Street
Newcastle-upon-Tyne NE1 6SU (GB)

(54) METHOD AND SYSTEM FOR PERFORMING SSL HANDSHAKE

(57) The present disclosure relates to the technical field of wireless communications and provides a method and system for performing an SSL Handshake. In the method, during an SSL handshake with a target terminal, a target CDN node determines a target service server accessed by the target terminal and obtains information to be processed by a private key; the target CDN node sends a private key processing request to a private key server corresponding to the target service server, the private key processing request carries the information to be processed and target private key processing type information; the private key server processes the information to be processed based on the target private key processing type information and a private key of the target service server and sends a processing result to the target CDN node so that the target CDN node may continue to perform the SSL handshake with the target terminal according to the processing result. By employing the present application, the security of the private key and the service quality of business service and CDN service may be guaranteed.

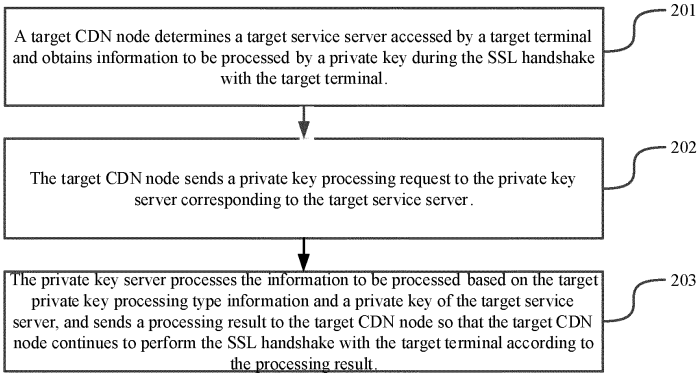

FIG. 2

EP 3 633 949 A1

Description

TECHNICAL FIELD

[0001] The present disclosure relates to the field of wireless communication technology, in particular, to a method and system for performing an SSL handshake.

BACKGROUND

[0002] With the rapid development of the Internet, requirements for network transmission security are increasingly higher, and Hyper Text Transfer Protocol over Secure Socket Layer (HTTPS) has therefore emerged. HTTPS may be deemed to be a combination of Hyper Text Transfer Protocol (HTTP) and the Secure Sockets Layer/Transport Layer Security (SSL/TLS) protocol. Herein, the SSL/TLS protocol works under the HTTP protocol and is used to encrypt transmitted data to ensure that the data is not intercepted/eavesdropped during transmission on the network.

[0003] Each service server can be pre-configured with an SSL certificate. The SSL certificate is issued by a trusted digital Certification Authority (CA), and granted to the service server after verifying the identity of the service server. The SSL certificate is used for server identity verification and data encryption. Each SSL certificate includes a public key and a private key. When a terminal accesses the service server through HTTPS, it needs to perform an SSL handshake with the service server firstly. During the SSL handshake, the terminal and the service server have to negotiate about an encryption algorithm to be used. If the encryption algorithm is the RSA algorithm, the service server has to decrypt an encrypted premaster password by using the private key of the SSL certificate. If the encryption algorithm is the DH algorithm, the service server has to sign a DH premaster by using the private key of the SSL certificate.

[0004] During implementation of the present disclosure, the inventor finds that at least the following problems exist in the existing technology.

[0005] Under the Content Delivery Network (CDN) service architecture, the terminal generally obtains data from a CDN node, thus the terminal has to perform the SSL handshake with the CDN node. However, on the one hand, due to the need for confidentiality, some service servers cannot provide the private key to the outside, so these service servers cannot use the CDN service. On the other hand, private key processing during the SSL handshake has to consume a large number of CPU processing resources, and especially, when there is a large amount of concurrency of the SSL handshake, the CDN service provided by the CDN node may be significantly affected.

SUMMARY

[0006] In order to solve the problems in the existing technology, embodiments of the present disclosure provide a method and system for performing an SSL handshake.

[0007] In a first aspect, it is provided a method for performing an SSL handshake, comprising:

determining, by a target CDN node, a target service server accessed by a target terminal and obtaining information to be processed by a private key during the SSL handshake with the target terminal;
sending, by the target CDN node, a private key processing request to a private key server corresponding to the target service server, the private key processing request carrying the information to be processed and target private key processing type information;
processing, by the private key server, the information to be processed based on the target private key processing type information and a private key of the target service server, and sending a processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result.

[0008] Alternatively, determining, by the target CDN node, the target service server accessed by the target terminal and obtaining information to be processed by the private key during the SSL handshake with the target terminal comprises:

determining, by the target CDN node, the target service server accessed by the target terminal according to a server identifier carried in a Client Hello message sent by the target terminal, and obtaining a premaster key carried in a Client Key Exchange message sent by the target terminal, where the premaster key is encrypted and is generated by the target terminal, during an RSA handshake with the target terminal under the TLS1.2 protocol;
sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the information to be processed and a target private key processing type, comprises:

sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the encrypted premaster key and an algorithm identifier of a target decryption algorithm negotiated with the target terminal;
processing, by the private key server, the information to be processed based on the target private key processing type and the private key of the target service server and sending the

processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result comprises:

decrypting, by the private key server, the encrypted premaster key by using the private key of the target service server according to the target decryption algorithm, and sending a premaster key obtained through decryption to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the premaster key obtained through the decryption.

[0009] Alternatively, determining, by the target CDN node, the target service server accessed by the target terminal and obtaining information to be processed by the private key during the SSL handshake with the target terminal comprises:

determining, by the target CDN node, the target service server accessed by the target terminal according to a server identifier carried in a client hello message sent by the target terminal, and obtaining a DH parameter generated locally, during a DH handshake with the target terminal under the TLS1.2 protocol;

sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the information to be processed and target private key processing type information, comprises:

sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the DH parameter and an algorithm identifier of a first signature algorithm negotiated with the target terminal;

processing, by the private key server, the information to be processed based on the target private key processing type information and the private key of the target service server and sending the processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result comprises:

signing, by the private key server, the DH parameter by using the private key of the target service server according to the first signature algorithm, and sending the signed DH parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed DH parameter.

[0010] Alternatively, determining, by the target CDN node, the target service server accessed by the target terminal and obtaining information to be processed by the private key during the SSL handshake with the target terminal comprises:

determining, by the target CDN node, the target service server accessed by the target terminal according to a server identifier carried in a client hello message sent by the target terminal, and obtaining a hash parameter generated locally, during a DH handshake with the target terminal under the TLS1.3 protocol, the hash parameter generated based on a handshake data packet and an SSL certificate of the target service server that interact with the target terminal;

sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the information to be processed and target private key processing type information, comprises:

sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the hash parameter and an algorithm identifier of a second signature algorithm negotiated with the target terminal;

processing, by the private key server, the information to be processed based on the target private key processing type information and the private key of the target service server and sending the processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result comprises:

signing, by the private key server, the hash parameter by using the private key of the target service server according to the second signature algorithm, and sending the signed hash parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed hash parameter.

[0011] Alternatively, the private key processing request further carries an SSL certificate identifier of the target service server;

before the private key server processes the information to be processed based on the target private key processing type information and the private key of the target service server, the method further comprises:

the private key server searches a private key list stored locally for a private key of the target service server corresponding to the SSL certificate identifier of the target service server.

[0012] Alternatively, the private key processing request is an http1.0 or http2.0 request.

[0013] Alternatively, sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server comprises:

determining, by the target CDN node, a next hop CDN node directed to the private key server corresponding to the target service server;
establishing an SSL bidirectional authentication channel between the target CDN node and the next hop CDN node, and sending the private key processing request to the next hop CDN node through the SSL bidirectional authentication channel so that the next hop CDN node sends the private key processing request to the private key server.

[0014] Alternatively, the private key server is deployed in a CDN service cluster or at the target service server.

[0015] In a second aspect, it is provided a system for performing an SSL handshake, comprising: a CDN node and a private key server.

[0016] A target CDN node is configured to determine a target service server accessed by a target terminal, obtain information to be processed by a private key during the SSL handshake with the target terminal, and to send a private key processing request to the private key server corresponding to the target service server, the private key processing request carries the information to be processed and target private key processing type information;

[0017] The private key server is configured to process the information to be processed based on the target private key processing type information and a private key of the target service server, and to send a processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result.

[0018] Alternatively, the target CDN node is specifically configured to:

determine the target service server accessed by the target terminal according to a server identifier carried in a client hello message sent by the target terminal, obtain a premaster key carried in a client key exchange message sent by the target terminal, where the premaster key is encrypted and is generated by the target terminal, during an RSA handshake with the target terminal under the TLS1.2 protocol, and send the private key processing request to the private key server corresponding to the target service server, the private key processing request carries the encrypted premaster key and an algorithm identifier of a target decryption algorithm negotiated with the target terminal;
the private key server is specifically configured to:
decrypt the encrypted premaster key by using the private key of the target service server according to the target decryption algorithm, and send a premaster key obtained through decryption to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the premaster key obtained through the decryption.

[0019] Alternatively, the target CDN node is specifically configured to:

determine the target service server accessed by the target terminal according to a server identifier carried in a client hello message sent by the target terminal and obtain a DH parameter generated locally during a DH handshake with the target terminal under the TLS1.2 protocol, and send the private key processing request to the private key server corresponding to the target service server, the private key processing request carries the DH parameter and an algorithm identifier of a first signature algorithm negotiated with the target terminal; and
the private key server is specifically configured to:
sign the DH parameter by using the private key of the target service server according to the first signature algorithm, and send the signed DH parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed DH parameter.

[0020] Alternatively, the target CDN node is specifically configured to:

determine the target service server accessed by the target terminal according to a server identifier carried in a client hello message sent by the target terminal, and obtain a hash parameter generated locally, during a DH handshake with the target terminal under the TLS1.3 protocol. Here the hash parameter is generated based on a handshake data packet and an SSL certificate of the target service server that interact with the target terminal. The target CDN node is further configured to send the private key processing request to the private key server corresponding to the target service server, the private key processing request carries the hash parameter and an algorithm identifier of a second signature algorithm negotiated with the target terminal; and
the private key server is specifically configured to:
sign the hash parameter by using the private key of the target service server according to the second signature algorithm, and send the signed hash parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed hash parameter.

[0021] Alternatively, the private key processing request further carries an SSL certificate identifier of the target service server; and
the private key server is further configured to:

search a private key list stored locally for a private key of the target service server corresponding to the SSL certificate identifier of the target service server.

[0022] Alternatively, the private key processing request is an http1.0 or http2.0 request.

[0023] Alternatively, the target CDN node is specifically configured to:

determine a next hop CDN node directed to a private key server corresponding to the target service server, establish an SSL bidirectional authentication channel between the target CDN node and the next hop CDN node, and send the private key processing request to the next hop CDN node through the SSL bidirectional authentication channel so that the next hop CDN node sends the private key processing request to the private key server.

[0024] Alternatively, the private key server is deployed in a CDN service cluster or at the target service server.

[0025] The technical solutions provided in the embodiments of the present disclosure result in the following advantageous effect:

In the embodiments of the present disclosure, a target CDN node determines a target service server accessed by a target terminal and obtains information to be processed by a private key during the SSL handshake with the target terminal; the target CDN node sends a private key processing request to a private key server corresponding to the target service server, the private key processing request carries the information to be processed and target private key processing type information; the private key server processes the information to be processed based on the target private key processing type information and a private key of the target service server, and sends a processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result. In this way, by constructing a private key server, the service server does not have to provide a private key to the outside, thereby ensuring security of the private key. Besides, a private key processing is completed by a private key server, so as to save CPU processing resources of a CDN node and a service server, thereby ensuring service quality of a CDN service and a business service.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] In order to illustrate the technical solutions in the embodiments of the present disclosure more clearly, the drawings used in the description of the embodiments will be briefly described below. It is obvious that the drawings in the following description are only some embodiments of the present disclosure. For those skilled in the art, other drawings may also be obtained according to the drawings without any inventive effort.

FIG. 1 is a schematic diagram of a network framework for performing an SSL handshake according to an embodiment of the present disclosure;

FIG. 2 is a method flowchart for performing the SSL handshake according to an embodiment of the present disclosure;

FIG. 3 is a method flowchart for performing an RSA handshake under the TLS1.2 protocol according to an embodiment of the present disclosure;

FIG. 4 is a TLS1.2 RSA handshake signaling diagram according to an embodiment of the present disclosure;

FIG. 5 is a method flowchart for performing a DH handshake under the TLS1.2 protocol according to an embodiment of the present disclosure;

FIG. 6 is a TLS1.2 DH handshake signaling diagram according to an embodiment of the present disclosure;

FIG. 7 is a method flowchart for performing a DH handshake under the TLS1.3 protocol according to an embodiment of the present disclosure;

FIG. 8 is a TLS1.3 DH handshake signaling diagram according to an embodiment of the present disclosure;

FIG. 9 is a schematic diagram of a network framework for performing an SSL handshake according to another embodiment of the present disclosure;

FIG. 10 is a schematic diagram of a network framework for performing an SSL handshake according to still another embodiment of the present disclosure.

DETAILED DESCRIPTION

[0027] In order to make the objective, the technical solutions and the advantages of the present disclosure clearer, the embodiments of the present disclosure will be further described in details with reference to the accompany drawings.

[0028] An embodiment of the present disclosure provides a method for performing an SSL handshake. The method may be implemented by a CDN node server (hereinafter referred to as a CDN node) and a private key server together. Herein, the CDN node may be any node server in a CDN service cluster for receiving a data request from a user terminal. The CDN node may store a service data (such as a web page data and a resource file) provided by a service server, and may further store an SSL certificate of the service server and a corresponding public key. The private key server may be a server deployed at the service server and is configured to store a private key of the service server and provide a private key processing function. A network framework of the above-described scenario is shown in FIG. 1. Both the above-described CDN node and private key server may include a processor, a memory and a transceiver. The processor may be configured to perform the SSL handshake as described in the following processing. The memory may be configured to store data required in the following processing and data generated. The transceiver may be configured to receive and send relevant data during the following processing.

[0029] In the following, the processing shown in FIG. 2 is described in detail with specific implementations which may be as follows.

[0030] In step 201, a target CDN node determines a target service server accessed by a target terminal and obtains information to be processed by a private key during the SSL handshake with the target terminal.

[0031] In implementation, when a user is intended to access a certain website or obtain a certain data resource through a terminal, the user may send, through a terminal, an access request of a corresponding service server (the target service server is taken as an example below for description). The CDN service cluster may receive the access request, and then select one CDN node (such as the target CDN node) to provide a business service to the terminal. Further, the CDN service cluster may feed back to the terminal an Internet Protocol (IP) address of the target CDN node. Afterwards, the terminal may first establish a Transmission Control Protocol (TCP) connection with the target CDN node based on the IP address of the target CDN node, and then perform the SSL handshake with the target CDN node via the TCP connection.

[0032] During the SSL handshake with the target terminal, the target CDN node may determine the target service server that the target terminal is intended to access through a server identifier carried in a Client Hello message sent by the target terminal. The server identifier may be domain name information or an IP address of the server. Meanwhile, the target CDN node may obtain the information to be processed by the private key.

[0033] In step 202, the target CDN node sends a private key processing request to the private key server corresponding to the target service server.

[0034] Herein, the private key processing request carries the information to be processed and target private key processing type information.

[0035] In implementation, after the target CDN node obtains the information to be processed by the private key , the target CDN node may first determine an IP address of the private key server corresponding to the target service server, and then send the above-described information to be processed to the private key server in a form of the private key processing request and on the basis of the IP address, and indicate a corresponding private key processing type (i.e., carrying the target private key processing type information).

[0036] Alternatively, the private key processing request may be an http1.0 or http2.0 request.

[0037] In implementation, after the target CDN node obtains the information to be processed, an http1.0 or http2.0 request may be structured. The information to be processed and the target private key processing type information are added to the http1.0 or http2.0 request, and then the http1.0 or http2.0 request may be sent to the private key server. It shall be noted that if the http2.0 request is applied as the private key processing request, a multiplexing characteristic of the http2.0 request may be used, so as to reduce the number of connections between the target CDN node and other network devices during transmission. Further, a header compression function of the http2.0 request may be applied to reduce data transmission quantity.

[0038] Alternatively, the private key processing request may be transmitted through a CDN network, and an intelligent routing selection function of the CDN node is applied to accelerate transmission of the private key processing request. Correspondingly, the processing of step 202 may be as follows: the target CDN node determines a next hop CDN node directed to the private key server corresponding to the target service server; an SSL bidirectional authentication channel between the target CDN node and the next hop CDN node is established, and the private key processing request is sent to the next hop CDN node through the SSL bidirectional authentication channel so that the next hop CDN node sends the private key processing request to the private key server.

[0039] In implementation, when sending the private key processing request to the private key server, the target CDN node may perform the intelligent routing selection according to a network condition between nodes in the CDN service cluster, so as to determine the next hop CDN node directed to the private key server. Afterwards, the SSL bidirectional authentication channel between the target CDN node and the next hop CDN node that is selected is established, and the private key processing request is sent to the next hop CDN node through the SSL bidirectional authentication channel. Further, a following CDN node (including the above-described next hop CDN node) may refer to the above-described processing, and each following CDN node sends a private key processing request through the next hop CDN node selected by intelligent routing, until the private key processing request reaches the private key server. In this way, on the one hand, transmission of the private key processing request may be accelerated through the CDN service cluster. On the other hand, the SSL bidirectional authentication channel may prevent a service provided by the private server from being called by an untrusted network device or prevent the private key server from being masqueraded by an untrusted network device.

[0040] In step 203, the private key server processes the information to be processed based on the target private key processing type information and a private key of the target service server, and sends a processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result.

[0041] In implementation, the private key server may receive the private key processing request sent by the target CDN node and extract the information to be processed and the target private key processing type information carried in the private key processing request. Afterwards, the private key server may read a locally stored private key of the target service server, and then process the information to be processed by using the private key

on the basis of the target private key processing type information. Further, the private key server may feed back a result of the processing to the target CDN node. In this way, the target CDN node may continue to perform the SSL handshake with the target terminal according to the processing result after receiving the above-described processing result. It shall be noted that on the basis of the above-described CDN service cluster and the processing of the SSL bidirectional authentication channel transmitting the private key processing request, the private key server in this step may feed back the processing result to the target CDN node through a path selected by the intelligent routing and on the basis of the SSL bidirectional authentication channel that is established already.

[0042] Alternatively, the private server may store private keys of a plurality of service servers, thus the private processing request may further carry an SSL certificate identifier of the target service server. Correspondingly, there may be a processing before step 203 as follows: the private key server searches a private key list stored locally for the private key of the target service server corresponding to the SSL certificate identifier of the target service server.

[0043] In implementation, the private key server may be provided to a plurality of service servers to perform private key processing. Therefore, the private key server may maintain a private key list which records a corresponding relationship between SSL certificate identifiers of the service server and private keys of the service server. In this way, the private key server, after receiving the private key processing request sent by the target CDN node, may obtain the SSL certificate identifier of the target service server carried in the private key processing request. Further, the private key server may search the private key list stored locally for the private key of the target service server corresponding to the SSL certificate identifier of the target service server.

[0044] In combination with different versions of handshake procedures, the processing from step 201 to step 203 is described in detail as follows.

I. An RSA handshake under the TLS1.2 protocol. As shown in FIG. 3:

[0045] In step 301, a target CDN node determines a target service server accessed by a target terminal according to a server identifier carried in a Client Hello message sent by the target terminal, and obtains a premaster key carried in a Client Key Exchange message sent by the target terminal and the premaster key is encrypted and is generated by the target terminal, during an RSA handshake with the target terminal under the TLS1.2 protocol.

[0046] In step 302, the target CDN node sends the private key processing request to the private key server corresponding to the target service server.

[0047] Herein, the private key processing request carries the encrypted premaster key and an algorithm identifier of a target decryption algorithm negotiated with the target terminal.

[0048] In step 303, the private key server decrypts the encrypted premaster key by using the private key of the target service server according to the target decryption algorithm, and sends a premaster key obtained through decryption to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the premaster key obtained through the decryption.

[0049] In implementation, FIG. 4 is a TLS1.2 RSA handshake signaling diagram, a content of which may specifically be as follows.

1. A target terminal sends a Client Hello message to a target CDN node. The Client Hello message carries a SSL protocol version supported by the terminal, an algorithm suite list, an SSL extended data (containing a server identifier of a target service server), and a client random number.
2. After receiving the Client Hello message, the target CDN node determines the target service server accessed by the target terminal according to the server identifier carried in the Client Hello message.
3. The target CDN node returns a Server Hello message to the target terminal. The Server Hello message carries a selected SSL protocol version (which is TLS1.2 herein), a selected algorithm suite (containing the target decryption algorithm and a generation algorithm of a session key) and a Server random number.
4. The target CDN node sends a Certificate message to the target terminal. The Certificate message carries an SSL certificate of the target service server and a corresponding certificate chain (for verifying identity of the SSL certificate). The SSL certificate contains a public key.
5. The target CDN node sends a Server Hello Done message to the target terminal, which indicates completion of the processing of the target CDN node and waits for response of the target terminal.
6. The target terminal generates a premaster key and uses the public key to encrypt the premaster key.
7. The target terminal sends the Client Key Exchange message to the target CDN node. The Client Key Exchange message carries an encrypted premaster key.
8. The target terminal generates a session key through cooperation of the premaster key with the server random number and the client random number on the basis of a generation algorithm of a selected session key.
9. The target terminal sends a Change Cipher Spec message to the target CDN node, and then sends a Finished message.
10. The target CDN node obtains the encrypted premaster key carried in the Client Key Exchange mes-

sage after receiving the Client Key Exchange message.

11. The target CDN node sends a private key processing request to the private key server corresponding to the target service server through a CDN network. The private key processing request carries the encrypted premaster key and an algorithm identifier of the target decryption algorithm.

12. The private key server decrypts the encrypted premaster key by using the private key of the target service server according to the target decryption algorithm.

13. The private key server sends a premaster key obtained through decryption to the target CDN node.

14. The target CDN node obtains the premaster key, and generates a session key through cooperation of the premaster key with the server random number and the client random number on the basis of the generation algorithm of the selected session key.

15. The target CDN node sends the Change Cipher Spec to the target terminal, indicating completion of key negotiation and that the following messages will be encrypted by using the session key.

16. The target CDN node sends the Finished message to the target terminal and completes the SSL handshake with the target terminal.

[0050] It shall be noted that the above-described processing is also applicable to an RSA handshake procedure under a protocol such as the SSL3 protocol, the TLS1.0 protocol and the TLS1.1 protocol.

II. A DH handshake under the TLS1.2 protocol. As shown in FIG. 5:

[0051] In step 501, a target CDN node determines a target service server accessed by a target terminal according to a server identifier carried in a Client Hello message sent by the target terminal, and obtains a DH parameter generated locally, during a DH handshake with the target terminal under the TLS1.2 protocol.

[0052] In step 502, the target CDN node sends a private key processing request to a private key server corresponding to the target service server.

[0053] Herein, the private key processing request carries the DH premaster and an algorithm identifier of a first signature algorithm negotiated with the target terminal.

[0054] In step 503, the private key server signs the DH parameter by using the private key of the target service server according to the first signature algorithm, and sends a signed DH parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed DH parameter.

[0055] In implementation, FIG. 6 is a TLS1.2 DH handshake signaling diagram, a content of which may specifically be as follows.

1. A target terminal sends a Client Hello message to a target CDN node. The Client Hello message carries a SSL protocol version supported by the terminal, an algorithm suite list, and an SSL extended data (containing a server identifier of a target service server), and a Client random number.

2. After receiving the Client Hello message, the target CDN node determines a target service server accessed by the target terminal according to the server identifier carried in the Client Hello message.

3. The target CDN node returns a Server Hello message to the target terminal. The Server Hello message carries a selected SSL protocol version (which is TLS1.2 herein), a selected algorithm suite (containing the first signature algorithm and a generation algorithm of a session key) and a Server random number.

4. The target CDN node sends a Certificate message to the target terminal. The Certificate message carries an SSL certificate of the target service server and a corresponding certificate chain (for verifying identity of the SSL certificate). The SSL certificate contains a public key.

5. The target CDN node generates a DH parameter.

6. The target CDN node sends a private key processing request to the private key server corresponding to the target service server through the CDN network. The private key processing request carries the DH parameter and an algorithm identifier of the first signature algorithm.

7. The private key server signs the DH parameter by using the private key of the target service server according to the first signature algorithm.

8. The private key server sends the signed DH parameter to the target CDN node.

9. The target CDN node obtains the signed DH parameter and sends a Server Key Exchange message to the target terminal. The Server Key Exchange message carries the signed DH parameter.

10. The target CDN node sends a Server Hello Done message to the target terminal, which indicates completion of the target CDN node and waits for response of the target terminal.

11. The target terminal obtains the signed DH premaster carried in the Server Key Exchange message after receiving the Server Key Exchange message.

12. The target terminal performs signature verification on the signed DH parameter by using the public key to generate a Client DH parameter.

13. The target terminal sends a Client Key Exchange message to the target CDN node. The Client Key Exchange message carries the Client DH parameter.

14. The target terminal generates a session key through cooperation of the Client DH parameter with the server random number and the client random number on the basis of a generation algorithm of a selected session key.

15. The target terminal sends a Change Cipher Spec message to the target CDN node, and then sends a Finished message.
16. The target CDN node obtains the Client DH parameter carried in the Client Key Exchange message after receiving the Client Key Exchange message.
17. The target CDN node generates a session key through cooperation of the Client DH parameter with the server random number and the client random number on the basis of the generation algorithm of the selected session key.
18. The target CDN node sends the Change Cipher Spec to the target terminal, indicating completion of key negotiation and that the following messages will be encrypted by using the session key.
19. The target CDN node sends a Finished message to the target terminal and completes the SSL handshake with the target terminal.

[0056] It shall be noted that the above-described processing is also applicable to a DH handshake procedure under a protocol such as the SSL3 protocol, the TLS1.0 protocol and the TLS1.1 protocol.

III. A DH handshake under the TLS1.3 protocol. As shown in FIG. 7:

[0057] In step 701, a target CDN node determines a target service server accessed by a target terminal according to a server identifier carried in a Client Hello message sent by the target terminal, and obtains a hash parameter generated locally, during a DH handshake with the target terminal under the TLS1.3 protocol.

[0058] Herein, the hash parameter is generated on the basis of a handshake data packet and an SSL certificate of the target service server that interact with the target terminal.

[0059] In step 702, the target CDN node sends a private key processing request to a private key server corresponding to the target service server.

[0060] Herein, the private key processing request carries the hash parameter and an algorithm identifier of a second signature algorithm negotiated with the target terminal.

[0061] In step 703, the private key server signs the hash parameter by using the private key of the target service server according to the second signature algorithm, and sends a signed hash parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed hash parameter.

[0062] In implementation, FIG. 8 is a TLS1.3 DH handshake signaling diagram, a content of which may specifically be as follows.

1. a target terminal sends a Client Hello message to a target CDN node. The Client Hello message carries a SSL protocol version supported by the terminal, an algorithm suite list, and an SSL expansion data (containing a server identifier of the target service server), a key_share extended data (containing a DH type supported by the terminal and a DH parameter), and a Client random number.
2. After receiving the Client Hello message, the target CDN node determines the target service server accessed by the target terminal according to the server identifier carried in the Client Hello message.
3. The target CDN node returns a Server Hello message to the target terminal. The Server Hello message carries a selected SSL protocol version (which is TLS1.3 herein), a selected algorithm suite (containing a second signature algorithm and a generation algorithm of a session key), a key share extended data (containing a selected DH type and a DH parameter), and a Server random number.
4. The target CDN node sends a Certificate message to the target terminal. The Certificate message carries the SSL certificate of the target service server and a corresponding certificate chain (for verifying identity of the SSL certificate). The SSL certificate containing the public key.
5. The CDN node generates a hash parameter on the basis of a handshake data packet and an SSL certificate of the target service server that interact with the target terminal.
6. The target CDN node sends a private key processing request to the private key server corresponding to the target service server through the CDN network. The private key processing request carries the hash parameter and the algorithm identifier of the second signature algorithm.
7. The private key server signs the hash parameter by using the private key of the target service server according to the second signature algorithm.
8. The private key server sends the signed hash parameter to the target CDN node.
9. The target CDN node obtains the signed hash parameter and sends a Certificate Verify message to the target terminal. The Certificate Verify message carries the signed hash parameter.
10. The target CDN node generates a session key through cooperation of the selected DH type and DH parameter with the server random number and the client random number on the basis of a generation algorithm of a selected session key.
11. The target CDN node sends a Finished message to the target terminal.
12. After receiving the Certificate Verify message, the target terminal obtains the signed hash parameter carried by the Certificate Verify message.
13. The target terminal performs signature verification on the signed hash parameter by using the public key.
14. The target terminal generates a session key through cooperation of the selected DH type and DH parameter with the server random number and the

client random number on the basis of the generation algorithm of the selected session key.

15. The target terminal sends a Finished message to the target CDN node and completes the SSL handshake with the target terminal.

[0063] In embodiments of the present disclosure, a target CDN node determines a target service server accessed by a target terminal and obtains information to be processed by a private key during the SSL handshake with the target terminal; the target CDN node sends a private key processing request to a private key server corresponding to the target service server, the private key processing request carrying the information to be processed and target private key processing type information; the private key server processes the information to be processed based on the target private key processing type information and a private key of the target service server, and sends a processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result. In this way, by constructing a private key server, the service server does not have to provide a private key to the outside, thereby ensuring security of the private key. Besides, a private key processing is completed by a private key server, so as to save CPU processing resources of a CDN node and a service server, thereby ensuring service quality of a CDN service and a business service.

[0064] In another embodiment, a private server may be a server that is deployed in a CDN service cluster, that is configured to store a private key of a service server, and that provides a function of private key processing. A specific network framework may be as shown in FIG. 9. Since private key processing in an SSL handshake may consume a large amount of CPU processing resources, and especially when there is a large amount of concurrency of the SSL handshake, a CDN service provided by a CDN node may be significantly affected. A private key processing cluster may be set up in the CDN service cluster. The private key processing cluster includes a large number of private key servers that may store private keys of a plurality of service servers. In this way, during an SSL handshake with a target terminal, a target CDN node may first determine a target service server accessed by the target terminal, and obtain information to be processed by a private key , and may then determine, in the private key processing cluster, a target private key server storing a private key of the target service server and send a private key processing request to the target private key server. The private key processing request may carry information to be processed and target private key processing type information. Further, the target private key server may process the information to be processed on the basis of the target private key processing type information and the private key of the target service server, and send a processing result to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result.

[0065] In another embodiment, apart from deploying a private server, some CDN nodes may also store a private key of a service server, and a specific network framework may be as shown in FIG. 10. During an SSL handshake with a terminal, after a CDN node sends a private key processing request to a private key server through a CDN network, if no processing result fed back by the private key server is received within a preset time period, the CDN node may perform private key processing by using a private key stored locally. Further, if no response is obtained after private key processing requests are sent multiple times continuously to the private key server, the CDN node may perform the private key processing by directly using the private key stored locally during the SSL handshake within a following preset time period. Further, before sending the private key processing request to the private server, the CDN node may further detect a current local load. If the current local load is greater than a preset threshold, the CDN node sends a private key processing request to the private key server. If the current local load is less than the preset threshold, the CDN node may perform the private key processing by using the private key stored locally.

[0066] In each private key server as described above, an SSL acceleration card may be applied to implement a relevant private key processing. The SSL acceleration card provides an interface. When a private key server calls the interface, a private key and information to be processed may be used as interface parameters to be transmitted to the SSL acceleration card.

[0067] On the basis of the same technical concept, an embodiment of the present disclosure further provides a system for performing an SSL handshake. The system includes a CDN node and a private key server.

[0068] A target CDN node is configured to determine a target service server accessed by a target terminal, obtain information to be processed by a private key during the SSL handshake with the target terminal, and send a private key processing request to the private key server corresponding to the target service server. The private key processing request carries the information to be processed and target private key processing type information.

[0069] The private key server is configured to process the information to be processed based on the target private key processing type information and a private key of the target service server, and send a processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result.

[0070] Alternatively, the target CDN node is specifically configured to:

determine the target service server accessed by the target terminal according to a server identifier carried in a Client Hello message sent by the target terminal, obtain a premaster key that is encrypted and that is generated by the target terminal, the premaster key carried in a Cli-

ent Key Exchange message sent by the target terminal, during an RSA handshake with the target terminal under the TLS1.2 protocol, and send the private key processing request to the private key server corresponding to the target service server. The private key processing request carries the encrypted premaster key and an algorithm identifier of a target decryption algorithm negotiated with the target terminal.

[0071] The private key server is specifically configured to:

decrypt the encrypted premaster key by using the private key of the target service server according to the target decryption algorithm, and send a premaster key obtained through decryption to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the premaster key obtained through the decryption.

[0072] Alternatively, the target CDN node is specifically configured to:

determine the target service server accessed by the target terminal according to a server identifier carried in a Client Hello message sent by the target terminal and obtain a DH parameter generated locally during a DH handshake with the target terminal under the TLS1.2 protocol, and send the private key processing request to the private key server corresponding to the target service server. The private key processing request carries the DH parameter and an algorithm identifier of a first signature algorithm negotiated with the target terminal.

[0073] The private key server is specifically configured to:

sign the DH parameter by using the private key of the target service server according to the first signature algorithm, and send the signed DH parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed DH parameter.

[0074] Alternatively, the target CDN node is specifically configured to:

determine the target service server accessed by the target terminal according to a server identifier carried in a Client Hello message sent by the target terminal, and obtain a hash parameter generated locally, during a DH handshake with the target terminal under the TLS1.3 protocol. The hash parameter is generated based on a handshake data packet and an SSL certificate of the target service server that interact with the target terminal. The target CDN node is configured to send the private key processing request to the private key server corresponding to the target service server, and the private key processing request carries the hash parameter and an algorithm identifier of a second signature algorithm negotiated with the target terminal.

[0075] The private key server is specifically configured to:

sign the hash parameter by using the private key of the target service server according to the second signature algorithm, and send the signed hash parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed hash parameter.

[0076] Alternatively, the private key processing request further carries an SSL certificate identifier of the target service server.

[0077] The private key server is further configured to: search a private key list stored locally for the private key of the target service server corresponding to the SSL certificate identifier of the target service server.

[0078] Alternatively, the private key processing request is an http1.0 or http2.0 request.

[0079] Alternatively, the target CDN node is specifically configured to:

determine a next hop CDN node directed to the private key server corresponding to the target service server, establish an SSL bidirectional authentication channel between the target CDN node and the next hop CDN node, and send the private key processing request to the next hop CDN node through the SSL bidirectional authentication channel so that the next hop CDN node sends the private key processing request to the private key server.

[0080] Alternatively, the private key server is deployed in a CDN service cluster or at the target service server.

[0081] In this embodiment of the present disclosure, a target CDN node determines a target service server accessed by a target terminal and obtains information to be processed by a private key during the SSL handshake with the target terminal; the target CDN node sends a private key processing request to a private key server corresponding to the target service server, the private key processing request carrying the information to be processed and target private key processing type information; the private key server processes the information to be processed based on the target private key processing type information and a private key of the target service server, and sends a processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result. In this way, by constructing a private key server, the service server does not have to provide a private key to the outside, thereby ensuring security of the private key. Besides, a private key processing is completed by a private key server, so as to save CPU processing resources of a CDN node and a service server, thereby ensuring service quality of a CDN service and a business service.

[0082] Those skilled in the art may appreciate that all or some steps that implement the above-described embodiments may be completed by hardware, or may be completed by a program that instructs relevant hardware. The program may be stored in a computer readable storage medium. The above-described storage medium may be a Read-Only Memory, a magnetic disc or an optical disc.

[0083] The above-described are only preferable embodiments of the present disclosure, but are not used to impose a limitation to the present disclosure. Any modi-

fication, equivalent substitution and improvement made within the spirit and principle of the present disclosure shall be included in the protection scope of the present disclosure.

Claims

1. A method for performing an SSL handshake, comprising:

   determining, by a target CDN node, a target service server accessed by a target terminal and obtaining, by the target CDN node, information to be processed by a private key during the SSL handshake with the target terminal;
   sending, by the target CDN node, a private key processing request to a private key server corresponding to the target service server, the private key processing request carrying the information to be processed and target private key processing type information;
   processing, by the private key server, the information to be processed based on the target private key processing type information and a private key of the target service server, and sending, by the private key server, a processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result.

2. The method according to claim 1, wherein determining, by the target CDN node, the target service server accessed by the target terminal and obtaining information to be processed by the private key during the SSL handshake with the target terminal comprises:

   determining, by the target CDN node, the target service server accessed by the target terminal according to a server identifier carried in a Client Hello message sent by the target terminal, and obtaining, by the target CDN node, a premaster key carried in a Client Key Exchange message sent by the target terminal, during an RSA handshake with the target terminal under the TLS1.2 protocol; wherein, the premaster key is encrypted and is generated by the target terminal;
   wherein sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the information to be processed and a target private key processing type information, comprises:

      sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the premaster key that is encrypted and an algorithm identifier of a target decryption algorithm negotiated with the target terminal; and
      wherein processing, by the private key server, the information to be processed based on the target private key processing type information and the private key of the target service server and sending the processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result comprises:
      decrypting, by the private key server, the premaster key that is encrypted by using the private key of the target service server according to the target decryption algorithm, and sending, by the private key server, a premaster key obtained through decryption to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the premaster key obtained through the decryption.

3. The method according to claim 1, wherein determining, by the target CDN node, the target service server accessed by the target terminal and obtaining information to be processed by the private key during the SSL handshake with the target terminal comprises:

   determining, by the target CDN node, the target service server accessed by the target terminal according to a server identifier carried in a client hello message sent by the target terminal, and obtaining, by the target CDN node, a DH parameter generated locally, during a DH handshake with the target terminal under the TLS1.2 protocol;
   wherein sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the information to be processed and target private key processing type information, comprises:

      sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the DH parameter and an algorithm identifier of a first signature algorithm negotiated with the target terminal; and
      wherein processing, by the private key serv-

er, the information to be processed based on the target private key processing type information and the private key of the target service server and sending the processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result comprises: signing, by the private key server, the DH parameter by using the private key of the target service server according to the first signature algorithm, and sending, by the private key server, the signed DH parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed DH parameter.

4. The method according to claim 1, wherein determining, by the target CDN node, the target service server accessed by the target terminal and obtaining information to be processed by the private key during the SSL handshake with the target terminal comprises:

determining, by the target CDN node, the target service server accessed by the target terminal according to a server identifier carried in a client hello message sent by the target terminal, and obtaining, by the target CDN node, a hash parameter generated locally, during a DH handshake with the target terminal under the TLS1.3 protocol; wherein the hash parameter is generated based on a handshake data packet and an SSL certificate of the target service server that interact with the target terminal;
wherein sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the information to be processed and target private key processing type information, comprises:

sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the hash parameter and an algorithm identifier of a second signature algorithm negotiated with the target terminal; and
wherein processing, by the private key server, the information to be processed based on the target private key processing type information and the private key of the target service server and sending the processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result comprises: signing, by the private key server, the hash parameter by using the private key of the target service server according to the second signature algorithm, and sending, by the private key server, the signed hash parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed hash parameter.

5. The method according to claim 1, wherein the private key processing request further carries an SSL certificate identifier of the target service server; and
wherein before the private key server processes the information to be processed based on the target private key processing type information and the private key of the target service server, the method further comprises:
the private key server searches a private key list stored locally for the private key of the target service server corresponding to the SSL certificate identifier of the target service server.

6. The method according to claim 1, wherein the private key processing request is an httpl.O or http2.0 request.

7. The method according to claim 1, wherein sending, by the target CDN node, the private key processing request to the private key server corresponding to the target service server comprises:

determining, by the target CDN node, a next hop CDN node directed to the private key server corresponding to the target service server;
establishing an SSL bidirectional authentication channel between the target CDN node and the next hop CDN node, and sending a private key processing request to the next hop CDN node through the SSL bidirectional authentication channel so that the next hop CDN node sends a private key processing request to the private key server.

8. The method according to any of claims 1-7, wherein the private key server is deployed in a CDN service cluster or at the target service server.

9. A system for performing an SSL handshake, comprising: a CDN node and a private key server, wherein:

a target CDN node is configured to determine a target service server accessed by a target terminal, obtain information to be processed by a private key during the SSL handshake with the

target terminal, and send a private key processing request to the private key server corresponding to the target service server, the private key processing request carries the information to be processed and target private key processing type information; and
the private key server is configured to process the information to be processed based on the target private key processing type information and a private key of the target service server, and send a processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result.

10. The system according to claim 9, wherein the target CDN node is specifically configured to:

determine the target service server accessed by the target terminal according to a server identifier carried in a client hello message sent by the target terminal, obtain a premaster key carried in a client key exchange message sent by the target terminal, during an RSA handshake with the target terminal under the TLS1.2 protocol, wherein the premaster key is encrypted and is generated by the target terminal; and
the target CDN node is configured to send the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the premaster key that is encrypted and an algorithm identifier of a target decryption algorithm negotiated with the target terminal;
the private key server is specifically configured to:
decrypt the premaster key that is encrypted by using the private key of the target service server according to the target decryption algorithm, and send a premaster key obtained through decryption to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the premaster key obtained through the decryption.

11. The system according to claim 9, wherein the target CDN node is specifically configured to:

determine the target service server accessed by the target terminal according to a server identifier carried in a client hello message sent by the target terminal and obtain a DH parameter generated locally during a DH handshake with the target terminal under the TLS1.2 protocol, and send the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the DH parameter and an algorithm identifier of a first signature algorithm negotiated with the target terminal; and
the private key server is specifically configured to:
sign the DH parameter by using the private key of the target service server according to the first signature algorithm, and send the signed DH parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed DH parameter.

12. The system according to claim 9, wherein the target CDN node is specifically configured to:

determine the target service server accessed by the target terminal according to a server identifier carried in a client hello message sent by the target terminal, and obtain a hash parameter generated locally, during a DH handshake with the target terminal under the TLS1.3 protocol, wherein the hash parameter is generated based on a handshake data packet and an SSL certificate of the target service server that interact with the target terminal;
the target CDN node is configured to send the private key processing request to the private key server corresponding to the target service server, the private key processing request carrying the hash parameter and an algorithm identifier of a second signature algorithm negotiated with the target terminal; and
the private key server is specifically configured to:
sign the hash parameter by using the private key of the target service server according to the second signature algorithm, and send the signed hash parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed hash parameter.

13. The system according to claim 9, wherein the private key processing request further carries an SSL certificate identifier of the target service server; and
the private key server is further configured to:
search a private key list stored locally for the private key of the target service server corresponding to the SSL certificate identifier of the target service server.

14. The system according to claim 9, wherein the private key processing request is an http1.0 or http2.0 request.

15. The system according to claim 9, wherein the target CDN node is specifically configured to:
determine a next hop CDN node directed to the private key server corresponding to the target service

server, establish an SSL bidirectional authentication channel between the target CDN node and the next hop CDN node, and send a private key processing request to the next hop CDN node through the SSL bidirectional authentication channel so that the next hop CDN node sends a private key processing request to the private key server.

16. The system according to any of claims 9-15, wherein the private key server is deployed in a CDN service cluster or at the target service server.

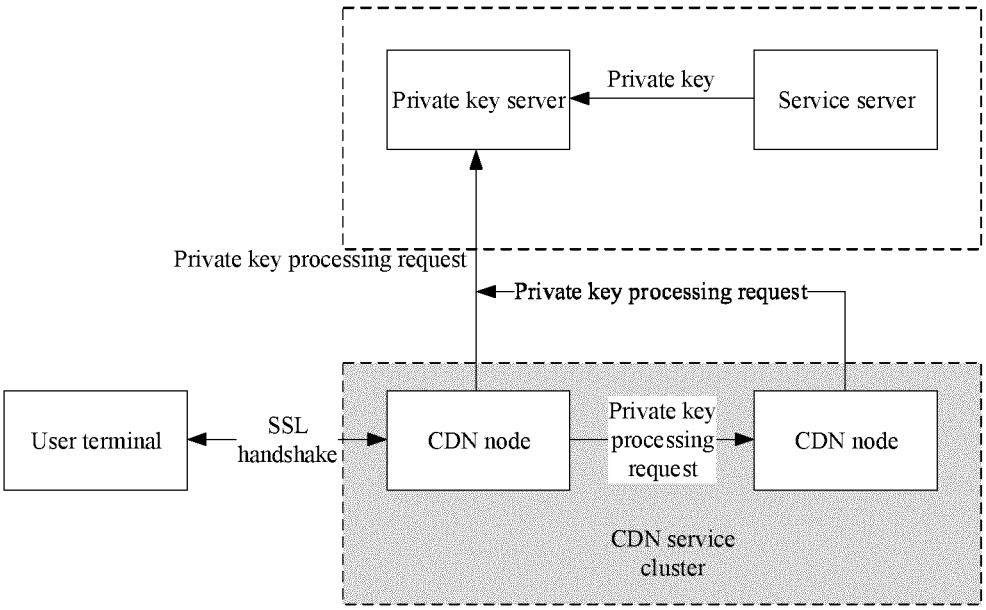
Private key server
Private key
Service server
Private key processing request
Private key processing request
User terminal
SSL handshake
CDN node
Private key processing request
CDN node
CDN service cluster

FIG. 1

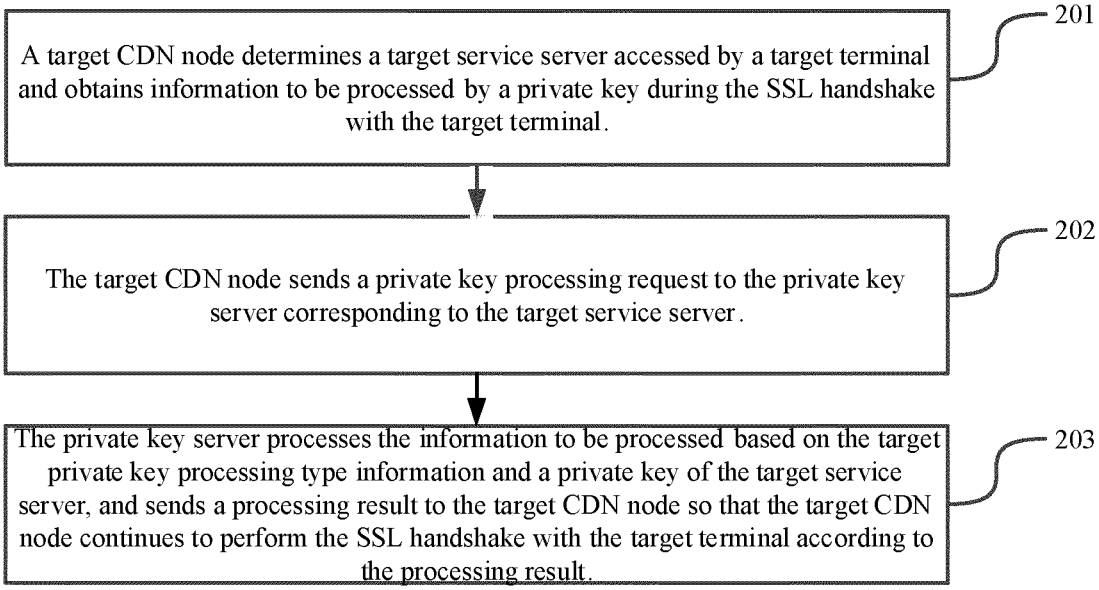
A target CDN node determines a target service server accessed by a target terminal and obtains information to be processed by a private key during the SSL handshake with the target terminal.
201
The target CDN node sends a private key processing request to the private key server corresponding to the target service server.
202
The private key server processes the information to be processed based on the target private key processing type information and a private key of the target service server, and sends a processing result to the target CDN node so that the target CDN node continues to perform the SSL handshake with the target terminal according to the processing result.
203

FIG. 2

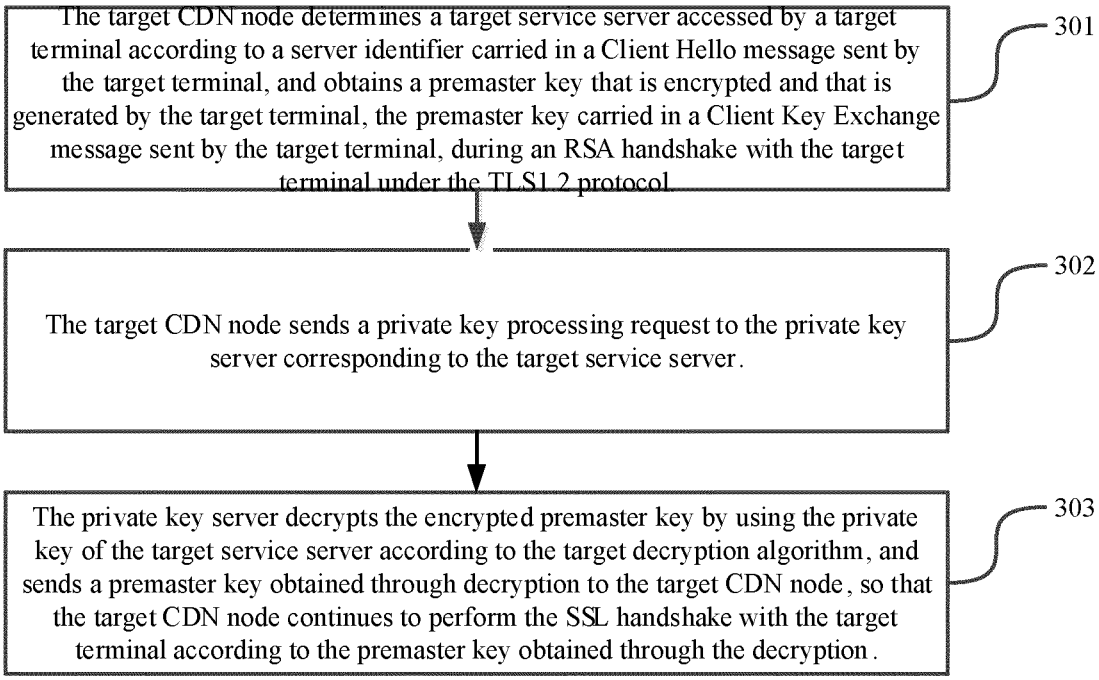
The target CDN node determines a target service server accessed by a target terminal according to a server identifier carried in a Client Hello message sent by the target terminal, and obtains a premaster key that is encrypted and that is generated by the target terminal, the premaster key carried in a Client Key Exchange message sent by the target terminal, during an RSA handshake with the target terminal under the TLS1.2 protocol
301
The target CDN node sends a private key processing request to the private key server corresponding to the target service server.
302
The private key server decrypts the encrypted premaster key by using the private key of the target service server according to the target decryption algorithm, and sends a premaster key obtained through decryption to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the premaster key obtained through the decryption.
303

FIG. 3

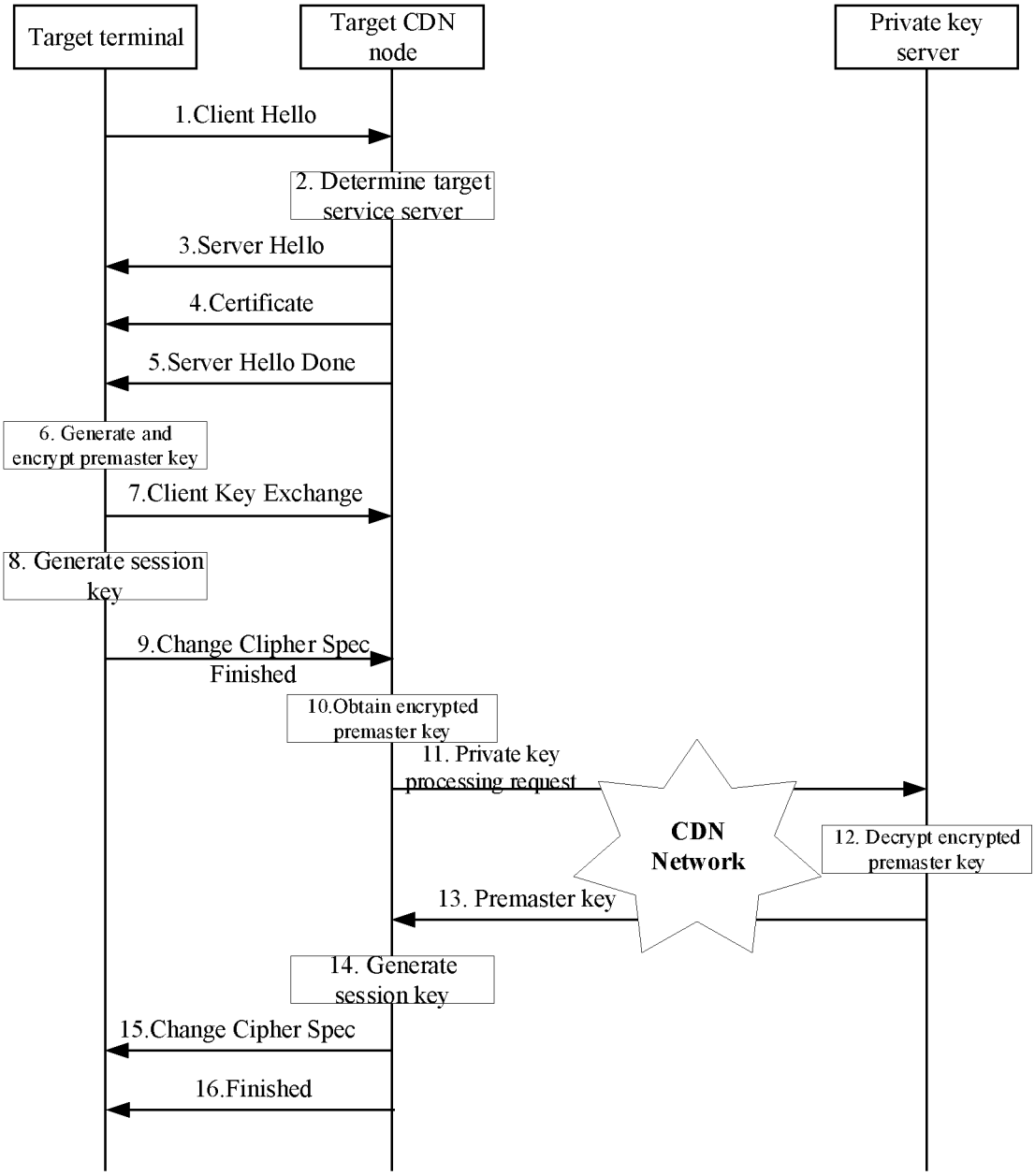
Target terminal
Target CDN node
Private key server
1.Client Hello
2. Determine target service server
3.Server Hello
4.Certificate
5.Server Hello Done
6. Generate and encrypt premaster key
7.Client Key Exchange
8. Generate session key
9.Change Clipher Spec
Finished
10.Obtain encrypted premaster key
11. Private key processing request
CDN Network
12. Decrypt encrypted premaster key
13. Premaster key
14. Generate session key
15.Change Cipher Spec
16.Finished

FIG. 4

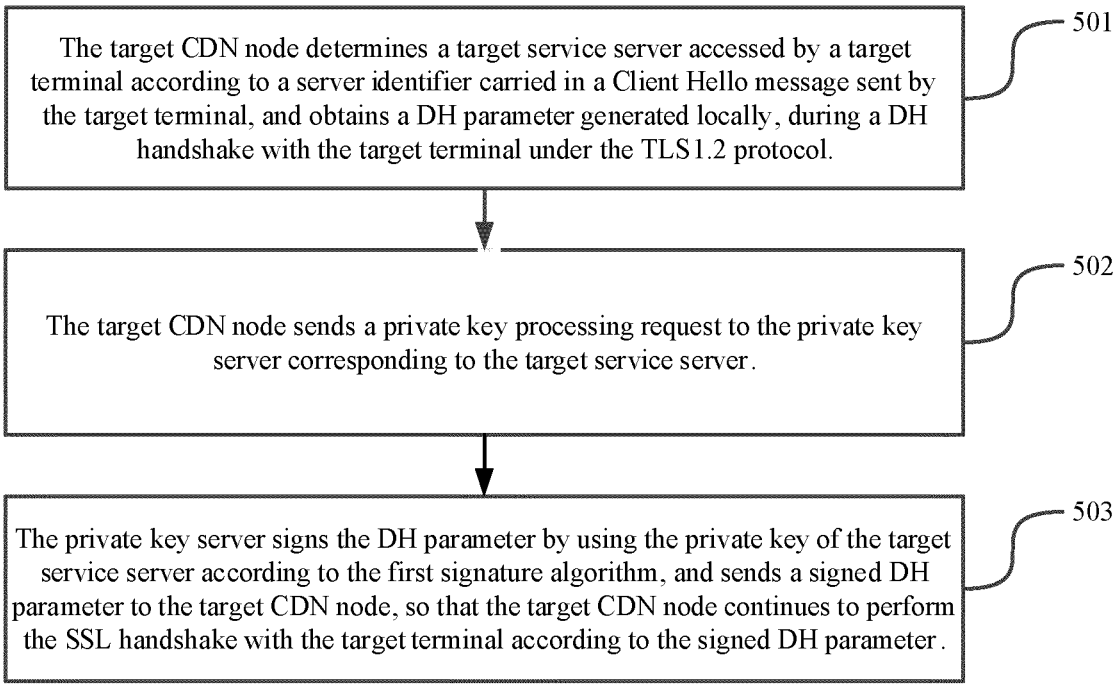
The target CDN node determines a target service server accessed by a target terminal according to a server identifier carried in a Client Hello message sent by the target terminal, and obtains a DH parameter generated locally, during a DH handshake with the target terminal under the TLS1.2 protocol.
501
The target CDN node sends a private key processing request to the private key server corresponding to the target service server.
502
The private key server signs the DH parameter by using the private key of the target service server according to the first signature algorithm, and sends a signed DH parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed DH parameter.
503

FIG. 5

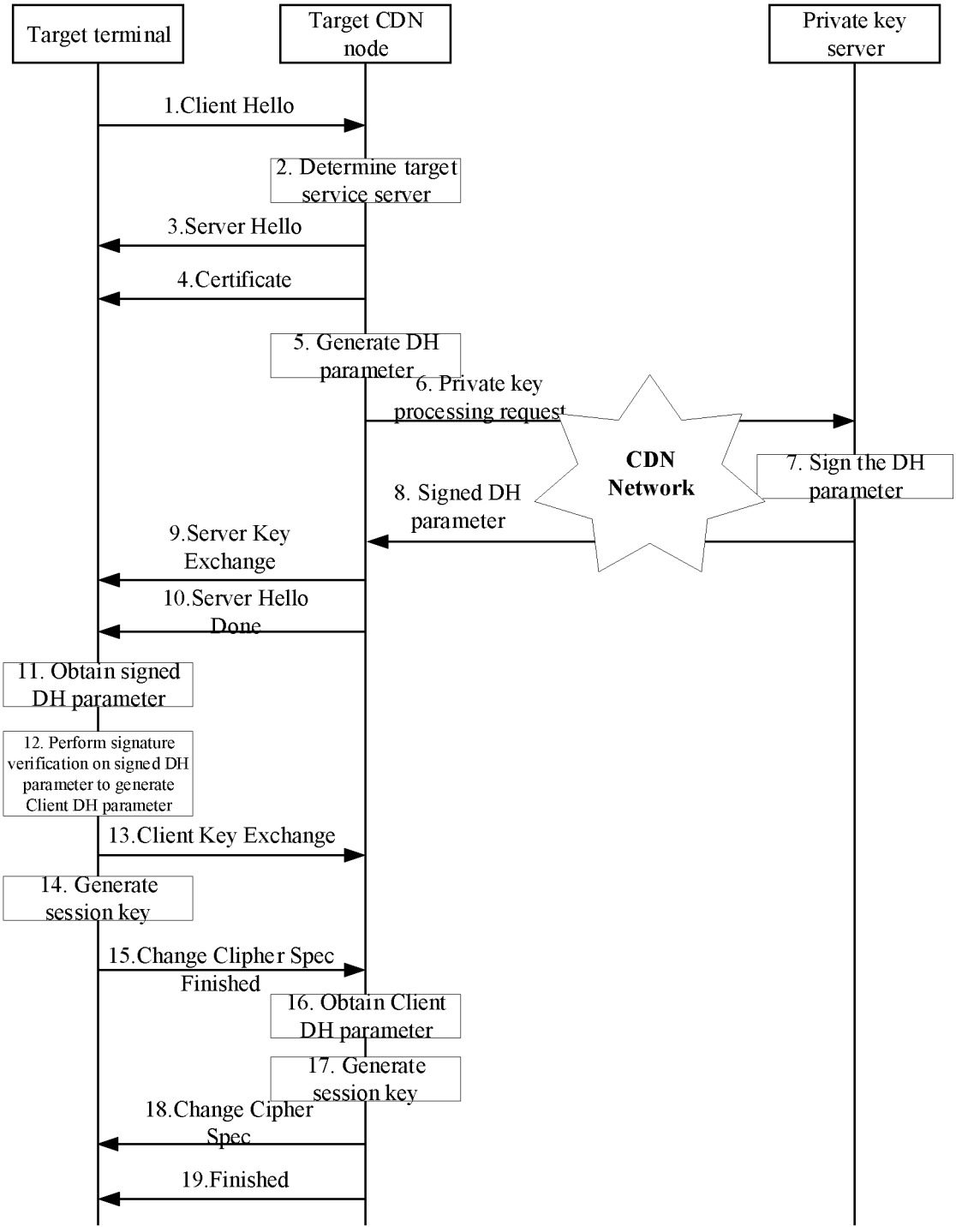
Target terminal
Target CDN node
Private key server
1.Client Hello
2. Determine target service server
3.Server Hello
4.Certificate
5. Generate DH parameter
6. Private key processing request
CDN Network
7. Sign the DH parameter
8. Signed DH parameter
9.Server Key Exchange
10.Server Hello Done
11. Obtain signed DH parameter
12. Perform signature verification on signed DH parameter to generate Client DH parameter
13.Client Key Exchange
14. Generate session key
15.Change Clipher Spec Finished
16. Obtain Client DH parameter
17. Generate session key
18.Change Cipher Spec
19.Finished

FIG. 6

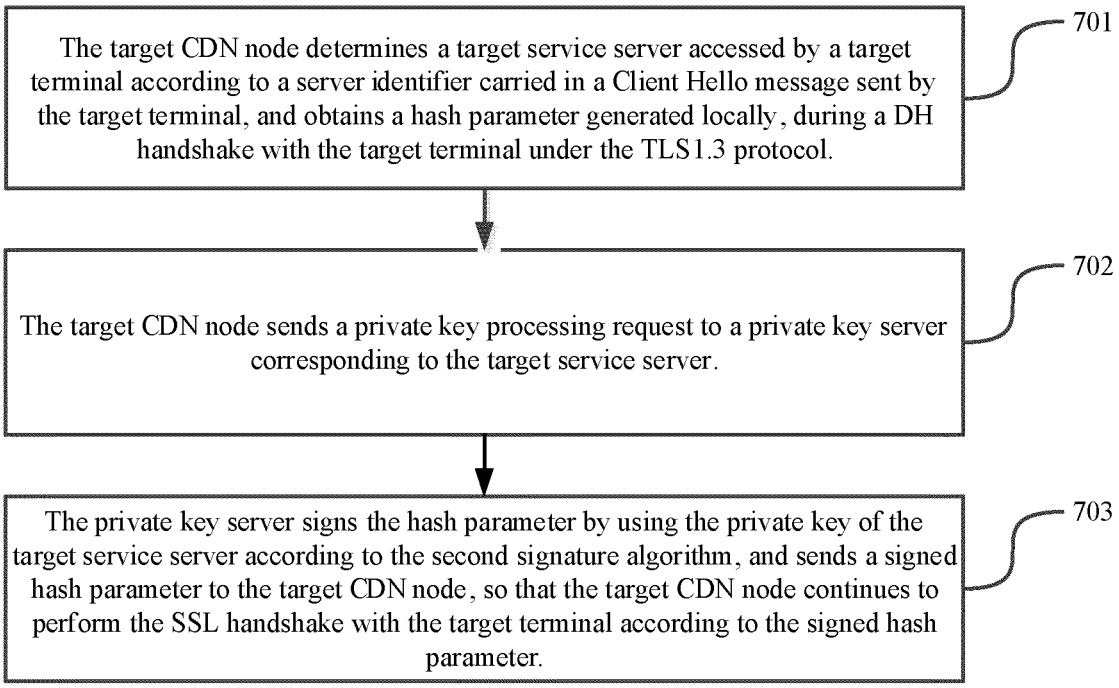
The target CDN node determines a target service server accessed by a target terminal according to a server identifier carried in a Client Hello message sent by the target terminal, and obtains a hash parameter generated locally, during a DH handshake with the target terminal under the TLS1.3 protocol.
701
The target CDN node sends a private key processing request to a private key server corresponding to the target service server.
702
The private key server signs the hash parameter by using the private key of the target service server according to the second signature algorithm, and sends a signed hash parameter to the target CDN node, so that the target CDN node continues to perform the SSL handshake with the target terminal according to the signed hash parameter.
703

FIG. 7

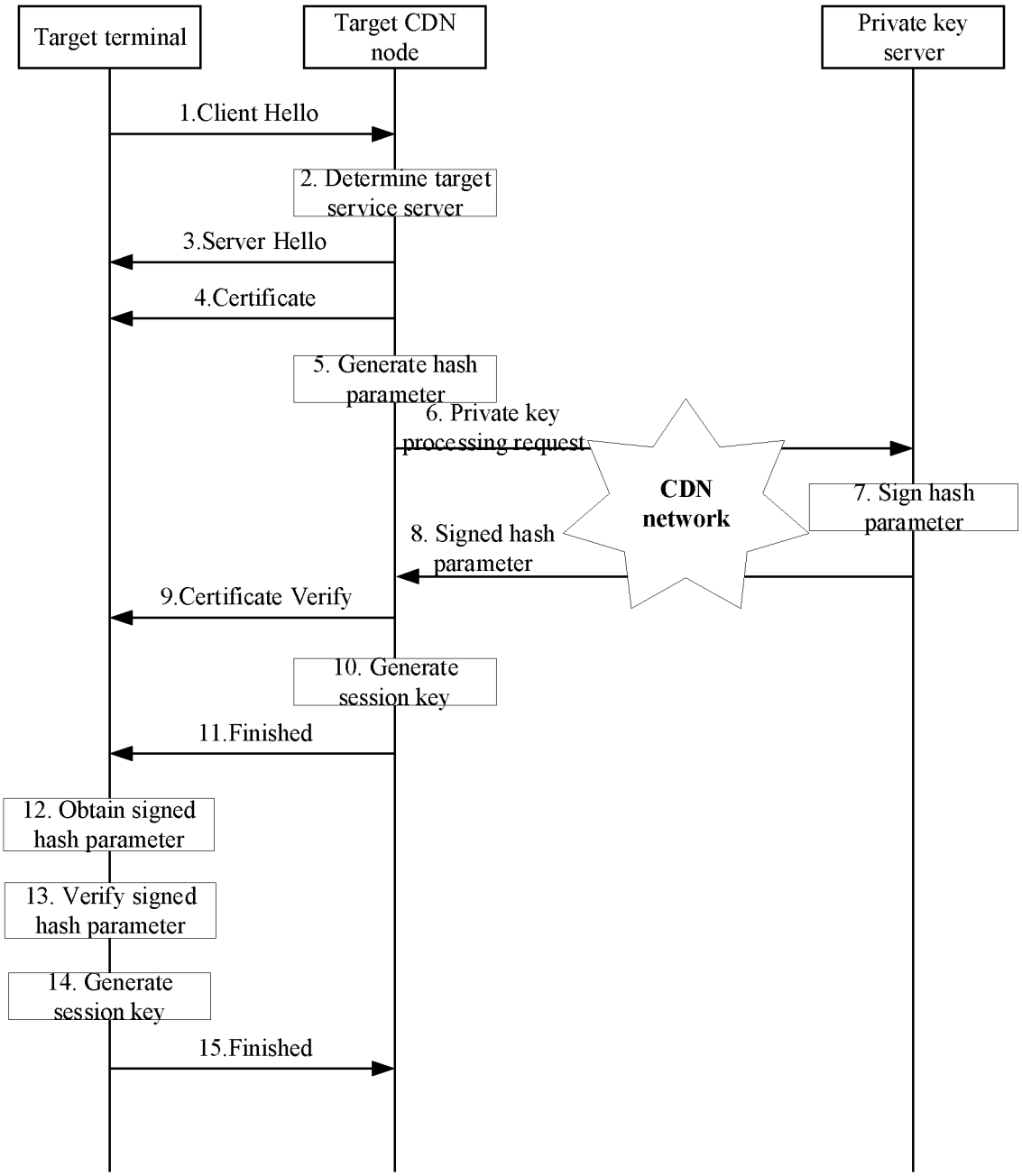
Target terminal
Target CDN node
Private key server
1.Client Hello
2. Determine target service server
3.Server Hello
4.Certificate
5. Generate hash parameter
6. Private key processing request
CDN network
7. Sign hash parameter
8. Signed hash parameter
9.Certificate Verify
10. Generate session key
11.Finished
12. Obtain signed hash parameter
13. Verify signed hash parameter
14. Generate session key
15.Finished

FIG. 8

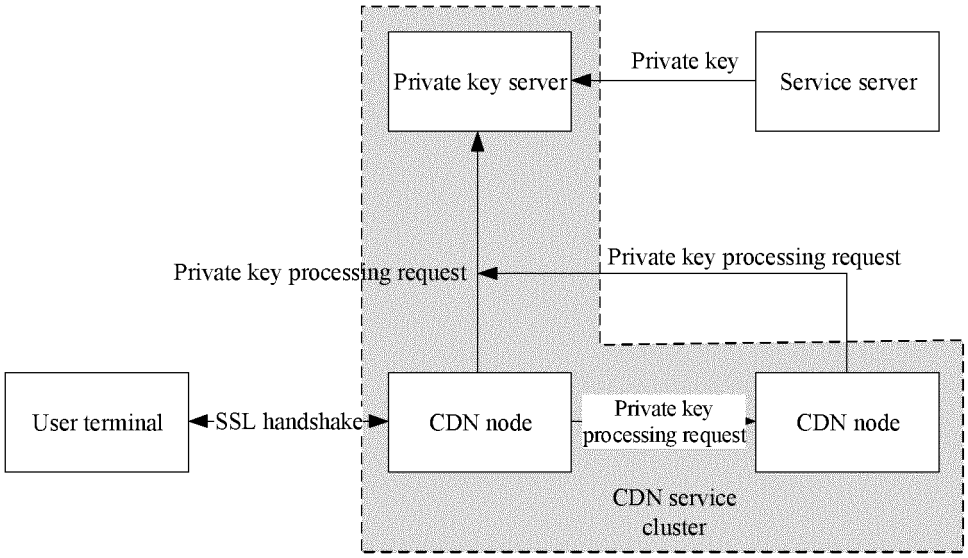
Private key server
Private key
Service server
Private key processing request
Private key processing request
User terminal
SSL handshake
CDN node
Private key processing request
CDN node
CDN service cluster

FIG. 9

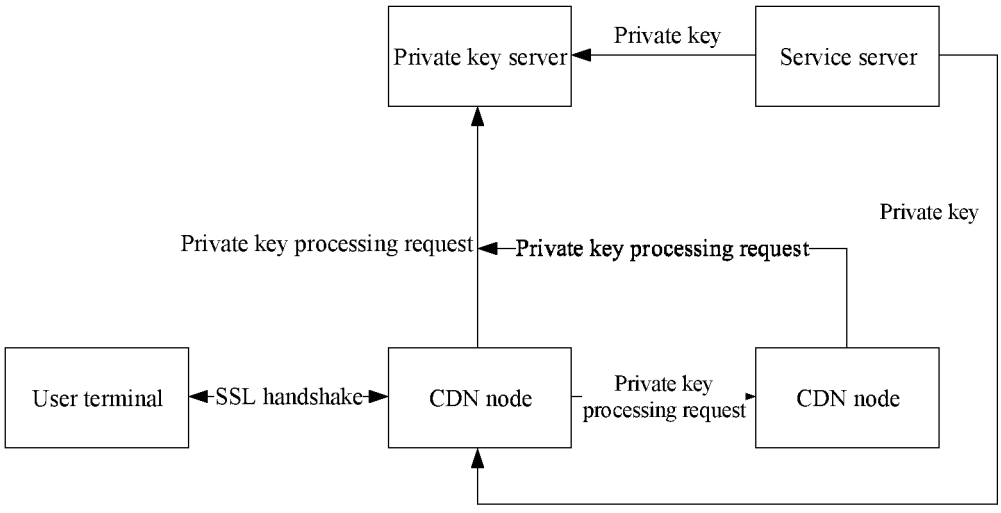
Private key server
Private key
Service server
Private key
Private key processing request
Private key processing request
User terminal
SSL handshake
CDN node
Private key processing request
CDN node

FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2018/088676

A. CLASSIFICATION OF SUBJECT MATTER

H04L 29/06(2006.01)i; H04L 29/08(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNKI, CNPAT, WPI, EPODOC: SSL, 握手, 秘钥, CDN, 服务器, handshake, server, key, encrypt+, decrypt+

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CN 106790090 A (BEIJING QIHOO TECHNOLOGY CO., LTD. ET AL.) 31 May 2017 (2017-05-31) description, paragraphs [0035]-[0063], and figures 1-3	1-16
A	CN 102801616 A (HUAWEI TECHNOLOGIES CO., LTD.) 28 November 2012 (2012-11-28) entire document	1-16
A	CN 105871797 A (LETV CLOUD COMPUTING CO., LTD.) 17 August 2016 (2016-08-17) entire document	1-16

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
26 November 2018	**29 December 2018**
Name and mailing address of the ISA/CN **State Intellectual Property Office of the P. R. China** **No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088** **China** Facsimile No. **(86-10)62019451**	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2018/088676

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 106790090 A	31 May 2017	None	
CN 102801616 A	28 November 2012	WO 2014019386 A1	06 February 2014
		US 2015156025 A1	04 June 2015
CN 105871797 A	17 August 2016	None	

Form PCT/ISA/210 (patent family annex) (January 2015)